Labor_01

LINEAR REGRESSION WITH ONE VARIABLE

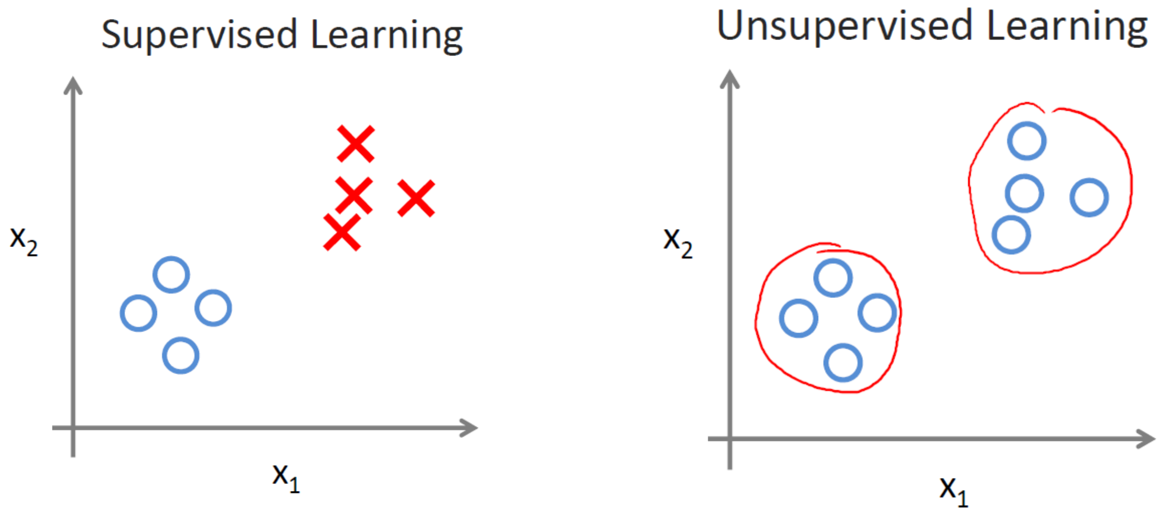
Machine Learning

„The field of study that gives computers the ability to learn without being explicitly programmed.”

/Arthur Samuel/

Main categories

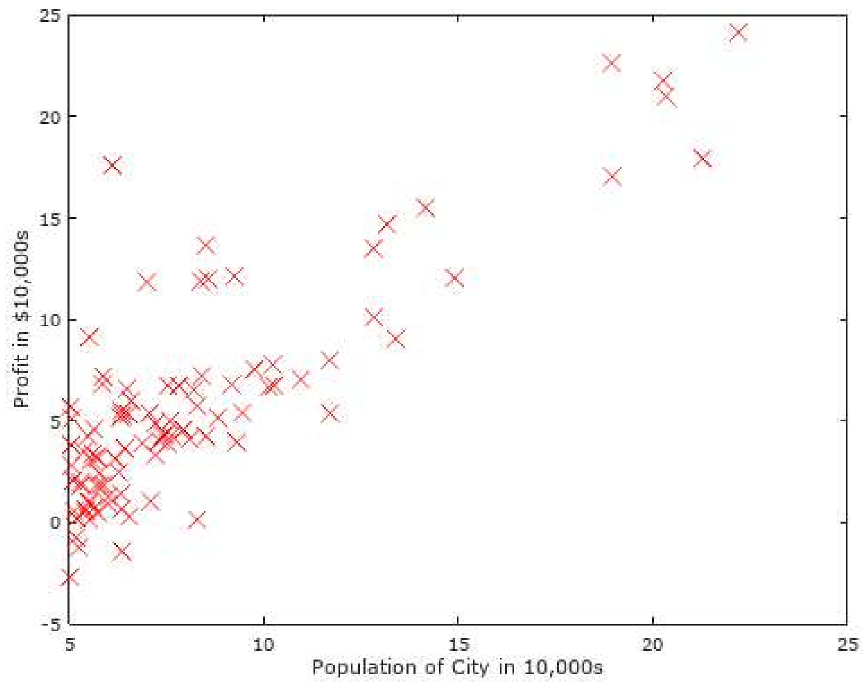
- Supervised learning
- Unsupervised learning



- Regression
- Classification

Linear Regression

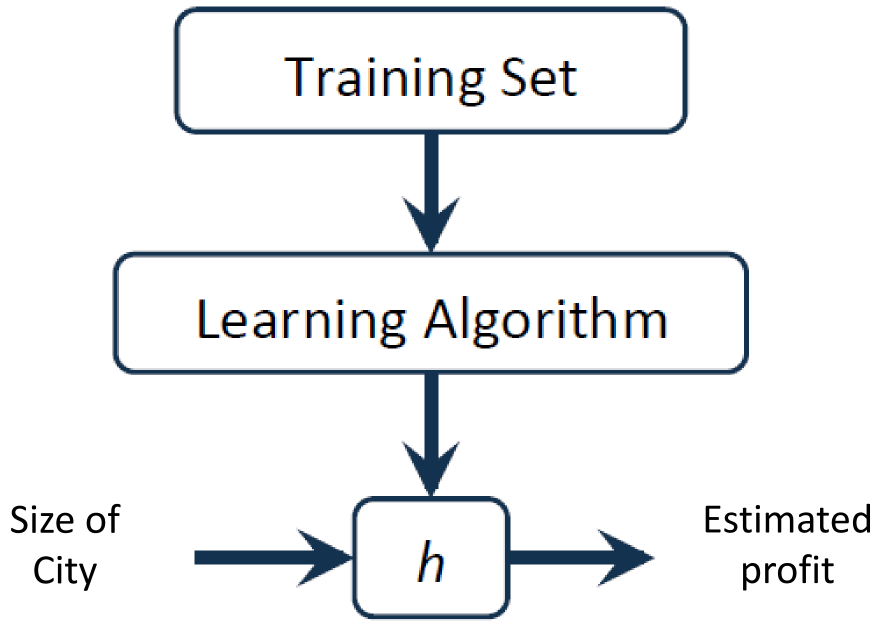
- One variable linear regression
  - You have data for profits and populations from different cities. You would like to use this data to help you select which city to expand your food truck company.



Model

Representation of h:

$$h_w(x) = w_0 + w_1 x$$



Cost function: MSE (Mean Squared Error)

$$C = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^i - y^i)^2$$

$$C = \frac{1}{2m}\sum_{i=1}^{m}(h_w(x^i) - y^i)^2$$

$$C(w_0, w_1)$$

$$C(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}(w_0 + w_1 x^i - y^i)^2$$

Data representations

Input data: X vector

Weights: W vector

Supervised output: Y vector

$$\underset{m\times 1}{X} = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ \vdots \\ x^m \end{bmatrix}, \underset{2\times 1}{W} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \underset{m\times 1}{Y} = \begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ \vdots \\ y^m \end{bmatrix}$$

ADD +1 (BIAS) to X vector

$$X = \begin{bmatrix} 1 & x^1 \\ 1 & x^2 \\ 1 & x^3 \\ \vdots & \vdots \\ 1 & x^m \end{bmatrix} \Rightarrow \underset{m\times 2}{X} = \begin{bmatrix} x_0^1 & x_1^1 \\ x_0^2 & x_1^2 \\ x_0^3 & x_1^3 \\ \vdots & \vdots \\ x_0^m & x_1^m \end{bmatrix}$$

$$\hat{y} = h_w(x) = w_0 + w_1 x^i = w_0 x_0^i + w_1 x_1^i$$

Forward step

$$\underset{2\times 1}{W} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$\underset{m\times 2}{X} = \begin{bmatrix} x_0^1 & x_1^1 \\ x_0^2 & x_1^2 \\ x_0^3 & x_1^3 \\ \vdots & \vdots \\ x_0^m & x_1^m \end{bmatrix} \quad \begin{bmatrix} w_0 x_0^1 + w_1 x_1^1 \\ w_0 x_0^2 + w_1 x_1^2 \\ w_0 x_0^3 + w_1 x_1^3 \\ \vdots \\ w_0 x_0^m + w_1 x_1^m \end{bmatrix} = \underset{m\times 1}{\hat{Y}} = XW$$

$$C = \frac{\sum (XW - Y)^2}{2m}$$

Data set => Hypothesis => Cost function

Minimize the Cost function!

Gradient descent

To solve: $minC(w_0, \ldots, w_n)$

Algorithm:

repeat until convergence {

$$w_j := w_j - \mu \frac{\partial}{\partial w_j} C(w_0, w_1)$$

}

Simultaneous update!!!

Linear Reg. + Grad. Descent

Linear Regression Model

$$h_w(x) = w_0 + w_1 x$$

$$\boxed{C(w_0, w_1)} = \frac{1}{2m}\sum_{i=1}^{m}(h_w(x^i) - y^i)^2$$

Gradient descent

repeat until convergence {

$$w_j := w_j - \mu \frac{\partial}{\partial w_j}\boxed{C(w_0, w_1)}$$

}

$$\frac{\partial}{\partial w_j}C(w_0, w_1) = \frac{\partial}{\partial w_j}\frac{1}{2m}\sum_{i=1}^{m}(h_w(x^i) - y^i)^2 = \frac{\partial}{\partial w_j}\frac{1}{2m}\sum_{i=1}^{m}(w_0 + w_1 x^i - y^i)^2$$

$$(j = 0) \qquad \frac{\partial}{\partial w_j}C(w_0, w_1) = \frac{1}{m}\sum_{i=1}^{m}(w_0 + w_1 x^i - y^i)\cdot 1 = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\cdot x_0^i$$

$$(j = 1) \qquad \frac{\partial}{\partial w_j}C(w_0, w_1) = \frac{1}{m}\sum_{i=1}^{m}(w_0 + w_1 x^i - y^i)\cdot x_1^i = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)\cdot x_1^i$$

„Batch” Gradient Descent

Each step of gradient descent uses all the training examples

$$w_0 = w_0 - \frac{\mu}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i)$$

$$w_1 = w_1 - \frac{\mu}{m}\sum_{i=1}^{m}(h_w(x^i) - y^i) \cdot x^i$$

$$w_0 = w_0 - \frac{\mu}{m} sum(X * w - Y)$$

$$w_1 = w_1 - \frac{\mu}{m} sum(X * w - Y). * X(:,2)$$